

MAU-MAF Support for OData based Android Applications

This document and code samples provided here is valid for product Sybase Unwired Platform (SUP) - Online Data Proxy (ODP), release version 2.1.3 and onwards.

This document includes the details and code snippets for provisioning meta-data containers in OData based Android applications.



Suhas Narasimhan
(Associate Developer, Android Platform, ODP, Mobile Platform IND)

TABLE OF CONTENTS

INTRODUCTION.....	3
APPLICATION PROPERTY WIZARD	3
Application Connection Property Wizard	4
Application Connection Template Wizard.....	5
MOBILE APPLICATION UNIT (MAU) SUPPORT	6
SAMPLE CODE SNIPPETS.....	7
Android ODP SDK 2.1.3.....	7
Android ODP SDK 2.2.....	8

INTRODUCTION

This document includes the details and code snippets for provisioning meta-data containers to android devices.

Application developers can download the Customization Resource Bundle, which is nothing but a JAR file that includes a manifest file of name and version properties using API mentioned in this document along with the sample code snippet.

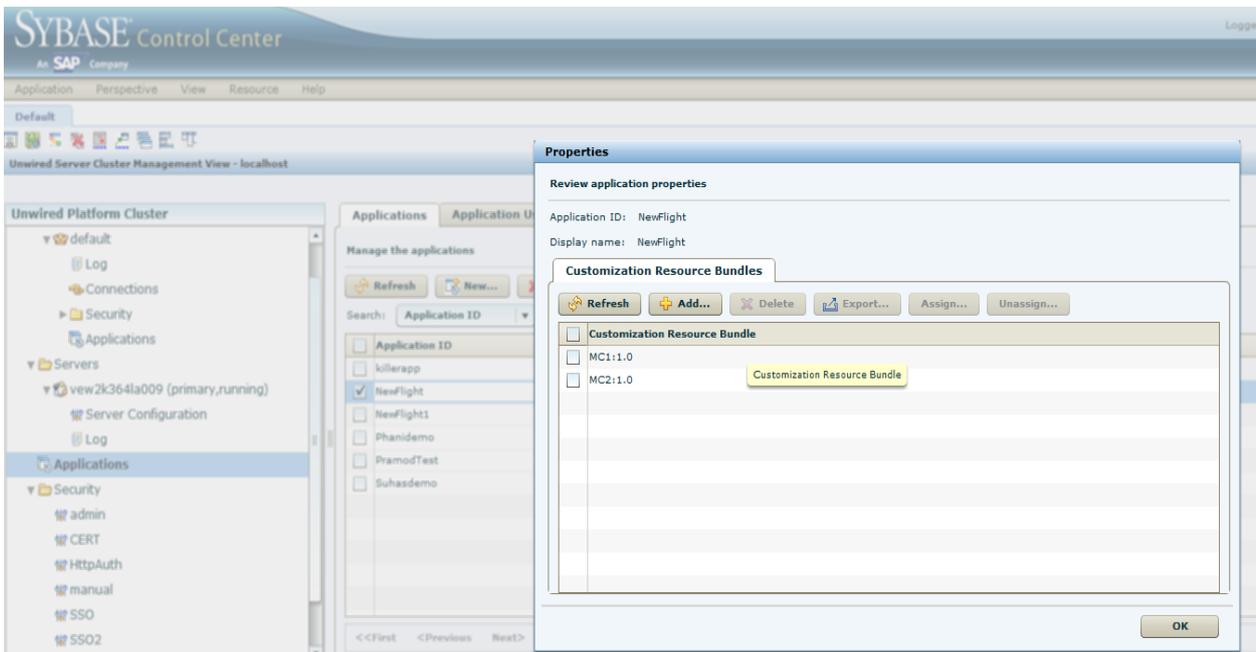
Customization resource bundles enable application developers to associate deployed client applications with different versions of customization resources. It does not contain any information that binds or helps bind to applications. It can be uploaded or exported during the definition of an application with Sybase Control Center(SCC).

Implementation of customization resource bundle requires the coordination of various roles:

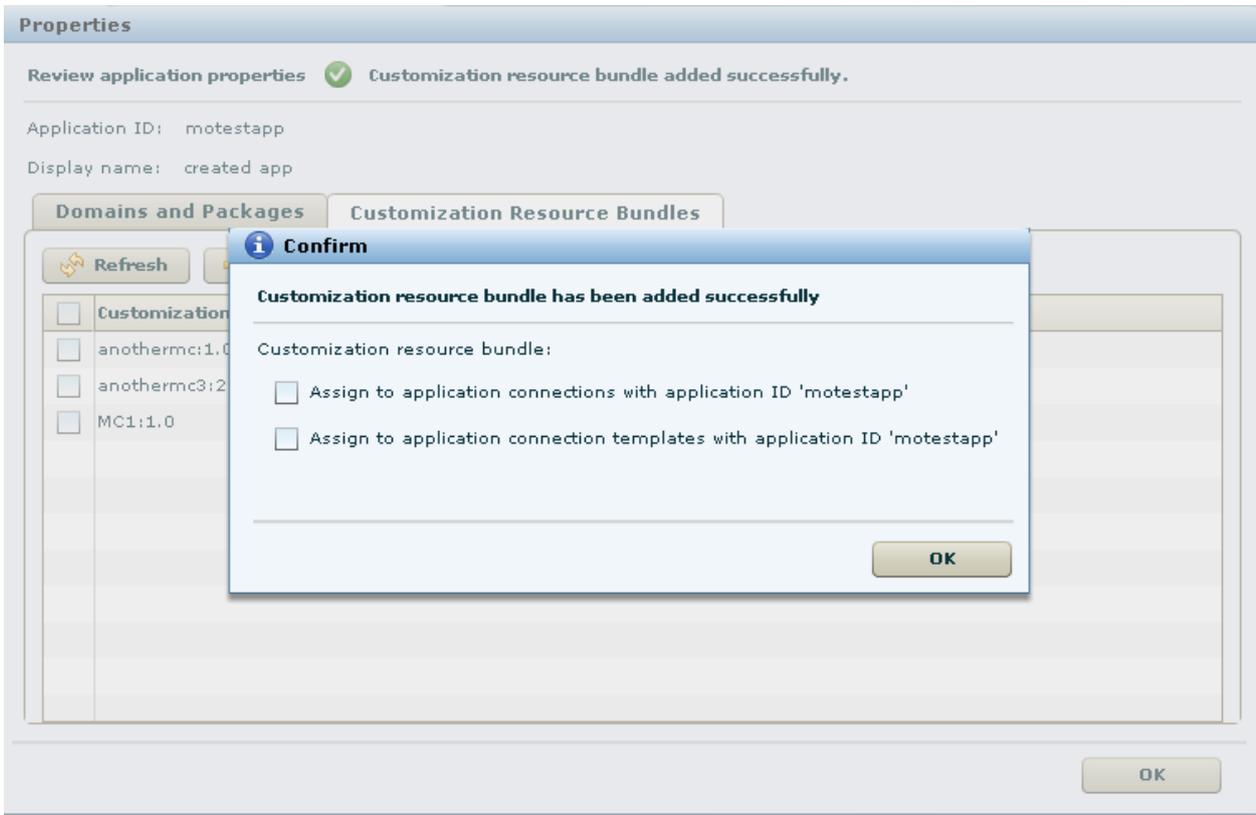
1. Application developer invokes the OData client SDK API that downloads the customization resource bundle using `getCustomizationResourceBundle()` method, which ties the application to the device and allows it to reach the client application.
2. Developer generates the JAR with the MANIFEST.MF, which includes these required properties:
 - Customization-Resource-Bundle-Name
 - Customization-Resource-Bundle-Version
3. Administrator then uploads the customization resource bundle to SUP Server and assigns it to an application connection or application connection template.

APPLICATION PROPERTY WIZARD

In SCC, under application properties application developer needs to add Customization Resource Bundle under Customization Resource bundle tab as shown below:

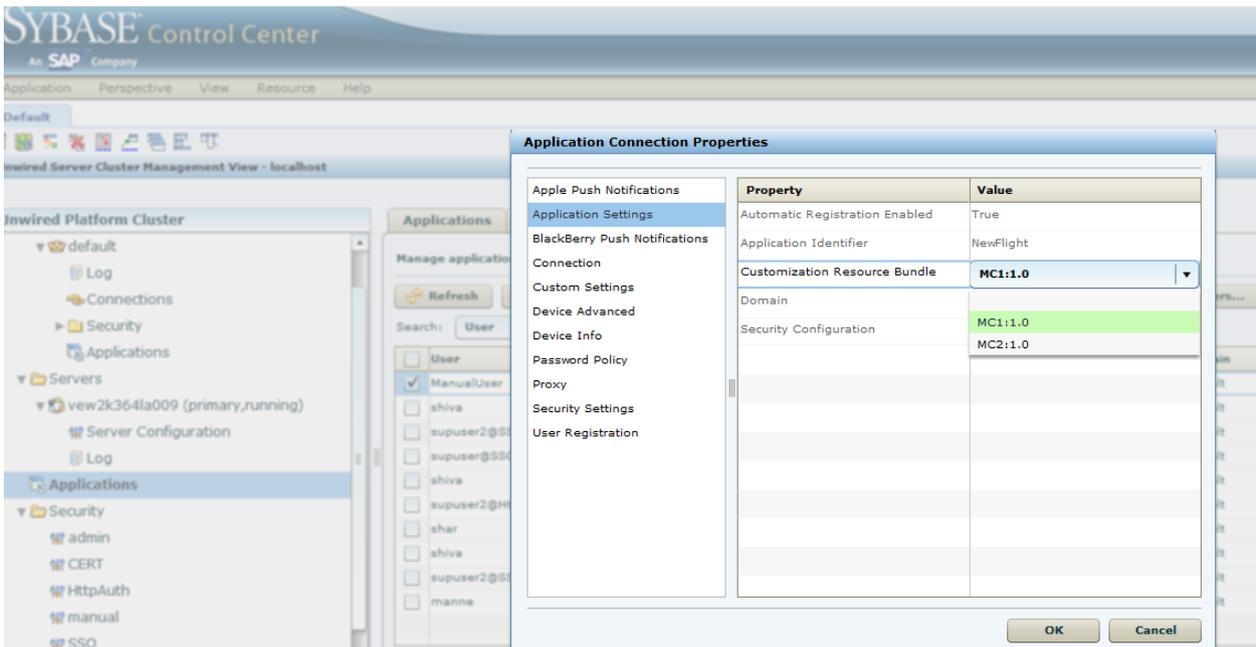


User selects the Customization Bundle zip/jar file here:



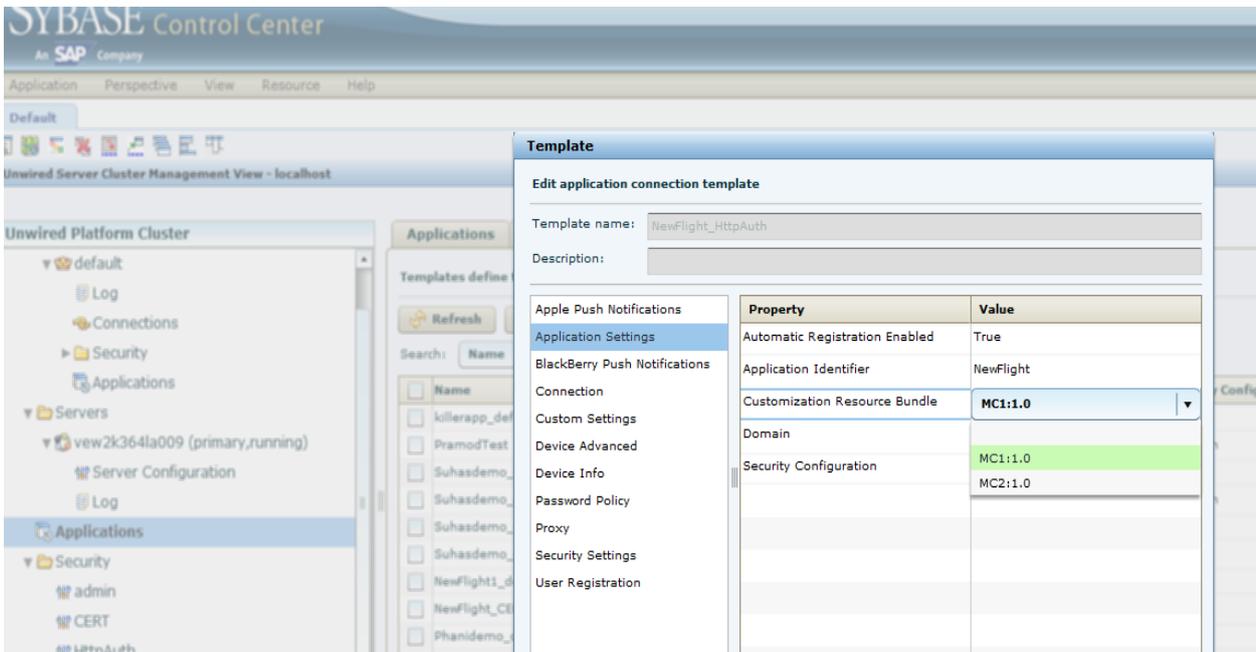
Application Connection Property Wizard

Once the Customization Bundle has been uploaded, one way to assign the Customization Resource Bundle is to select a particular user under Application Connections Tab. Once selected, developer can go to properties and assign the uploaded resource bundle under Customization Resource Bundle property in Application Settings as shown in below screenshot. So in this way application developer can assign different Customization Resource Bundles to different users.



Application Connection Template Wizard

Once the Customization Bundle has been uploaded, other way to assign the Customization Resource Bundle is to select an application Connection Template under Application Connections Tab. Once selected developer can go to properties and assign the uploaded resource bundle under Customization Resource Bundle property in Application Settings as shown in below screenshot. So in this way Resource Bundle assigned to that connection template can be reused for different users created from same template.



MOBILE APPLICATION UNIT (MAU) SUPPORT

Once Resource Bundle has been assigned to the application connection template , application developers can add below provided code snippet in their applications to receive the notification everytime there is any change in bundle assigned in SCC. It returns the Customization Resource in the form of binary data.

Method Signature – 2.1.3

```
/**
 * This method return/downloads the Customization Resources as binary data
 after it is notified by changes of "CUSTOMIZATION_RESOURCE" property (ID 2903)
 * @param CUSTOMIZATION_RESOURCE if null/empty fetches the default as
 assigned Customization Resource
 * @param UserName the User Name
 * @param Password the Password
 * @return the Customization Resource zip/jar file as binary data
 * @throws MessagingClientException
 */

public byte[] getCustomizationResourceBundle (String
CUSTOMIZATION_RESOURCE,String UserName , String Password)throws
MessagingClientException{
}
```

Method Signature – 2.2 and onwards

```
/**
 * This method return/downloads the Customization Resources as binary data
 after it is notified by changes of "CUSTOMIZATION_RESOURCE" property (ID 2903)
 * @param CUSTOMIZATION_RESOURCE if null/empty fetches the default as
 assigned Customization Resource
 * @param UserName the User Name
 * @param Password the Password
 * @return the Customization Resource zip/jar file as binary data
 * @throws ODPExceptioin
 */

public byte[] getCustomizationResourceBundle (String
CUSTOMIZATION_RESOURCE,String UserName , String Password)throws ODPException{
}
```

SAMPLE CODE SNIPPETS

Android ODP SDK 2.1.3

Sample Code:

1. Initialize and set the listener

```
LiteMessagingClient.initInstance(getApplicationContext(), appName);
LiteMessagingClient lm= LiteMessagingClient.getInstance();
EndPointListner epl=new EndPointListner();
lm.addConfigurationChangeListener(epl);
serviceDocPath = las.getApplicationEndPoint();
```

2. Class the implements the SUPLiteConfigurationChangeListener

```
public class EndPointListner implements SUPLiteConfigurationChangeListener{

    @Override
    public void configurationHasChanged(int arg0, String arg1) {
        Log.i("Configuration has changed", "Value: "+arg1);
        String Custom_resource = "";
        if(arg0 == CUSTOMIZATION_RESOURCES)
        {
            LiteAppSettings las = new LiteAppSettings();
            byte[] Property_metadata;
            try {
                Property_metadata =
                    las.getCustomizationResourceBundle(arg1,"supuser2","s3pu
                    ser");
            } catch (MessagingClientException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }
            catch (UnsupportedEncodingException e) {
                // TODO Auto-generated catch block
                e.printStackTrace();
            }

            Log.i("MDP","Customiation resources test"+Custom_resource );

        }
    }
}
```

Android ODP SDK 2.2

Sample Code:

1. Initialize and set the listener

```
ODPClientConnection.initInstance(getApplicationContext(), appName);
ODPClientConnection client= ODPClientConnection.getInstance();
configChangeListener listener=new configChangeListener();
client.addConfigurationChangeListener(listener);
```

2. Class that implements the IODPConfigurationChangeListener

```
public class configChangeListener implements IODPConfigurationChangeListener{

    public void onConfigurationChange(int key, String value) {
//Is called when there is a customization bundle added/changed from SCC
        if(key==CUSTOMIZATION_RESOURCES)
            {
                try {
                    ODPAppSettings las = new ODPAppSettings();
                    byte bundleData[] = null;
                    try {
                        bundleData =
las.getCustomizationResourceBundle(value,Helper.USERNAME, Helper.PASSWORD);
                    } catch (ODPException e) {
                        e.printStackTrace();
                    }

                    ZipInputStream zipStream = new ZipInputStream(new
ByteArrayInputStream(bundleData));
                    ZipEntry entry = null;
                    while ((entry = zipStream.getNextEntry()) != null)
                    {
                        String entryName = entry.getName();
                        zipStream.closeEntry();
                        Log.i("Tag",
"FileName***"+entryName+"size***"+entry.getCompressedSize());
                    }
                    zipStream.close();
                }
                catch (IOException e) {
                    throw new RuntimeException(e);
                }
            }
    }
}
```

© 2012 SAP AG. All rights reserved.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, SAP HANA, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company.

Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase Inc. Sybase is an SAP company.

Crossgate, m@gic EDDY, B2B 360°, and B2B 360° Services are registered trademarks of Crossgate AG in Germany and other countries. Crossgate is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

These materials are subject to change without notice. These materials are provided by SAP AG and its affiliated companies ("SAP Group") for informational purposes only, without representation or warranty of any kind, and SAP Group shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP Group products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

